

# **Software per a la generació d'arxius GCODE per a impressores 3D**

Autor:	Eloi Pardo Rivero
Director:	Eric Pallarés García
Ponent:	Albert Cabellos Aparicio
Especialitat:	Enginyeria del Software
Data defensa:	29 de Juny de 2018

# Resum

En els últims anys el món de la impressió 3D ha patit una pujada molt important, ja que s'ha vist que aquesta tecnologia es pot utilitzar per a moltes aplicacions, tant per a la indústria com per a l'ús quotidià.

Aquest projecte s'ha realitzat a BCN3D, una petita empresa que dissenya i fabrica impressores 3D. És una empresa amb pocs recursos i ha vist com en els darrers anys ha anat quedant enrere respecte la competència a nivell de *software*. Per a oferir una millor experiència als usuaris de les seves impressores volien poder oferir un *software* propi que aprofités totes les possibilitats de les seves impressores.

La solució que es desenvolupa en aquest projecte és la d'adaptar un *software open source* desenvolupat per *Ultimaker* per poder exprimir al màxim totes les funcionalitats de les impressores de BCN3D i donar més valor a la marca oferint un *software* propi.

Amb aquest nou *software*, els usuaris podran aprofitar la funcionalitat d'impressió duplicada, actualitzar-se a l'últim *firmware* de la impressora tan bon punt aquest hagi estat publicat, imprimir i controlar la impressora connectant-la per USB a un ordinador.

# Resumen

En los últimos años el mundo de la impresión 3D ha sufrido una subida muy importante, ya que se ha visto que esta tecnología se puede utilizar para muchas aplicaciones, tanto para la industria como para el uso cotidiano.

Este proyecto se ha realizado en BCN3D, una pequeña empresa que diseña y fabrica impresoras 3D. Es una empresa con pocos recursos y ha visto como en los últimos años se ha ido quedando atrás respecto la competencia a nivel de *software*. Para ofrecer una mejor experiencia a los usuarios de sus impresoras querían poder ofrece un *software* propio que aproveche todas las posibilidades de sus impresoras.

La solución que se desarrolla en este proyecto es la de adaptar un *software open source* desarrollado por *Ultimaker* para poder exprimir al máximo todas la funcionalidades de las impresoras de BCN3D y dar más valor a la marca ofreciendo un *software* propio.

Con esta nuevo *software*, los usuarios podrán aprovechar la funcionalidad de impresión duplicada, actualizarse al último *firmware* de la impresora cuando este se haya publicado, imprimir y controlar la impresora conectándola por USB a un ordenador.

# Abstract

In recent years the world of 3D printing has undergone a very important rise, as it has been seen that this technology can be used for many applications, both for industry and for everyday use.

This project has been done at BCN3D, a small company that designs and manufactures 3D printers. It is a company with few resources and in the last years has been behind the competitors with respect to *software*. To offer a better experience to the users of their printers they wanted to be able to offer their own *software* that took advantage of all the possibilities of their printers.

The solution that is developed in this project is the adaptation of an *open source software* developed by Ultimaker to be able to squeeze to the maximum all the functionalities of the BCN3D printers and give more value to the brand by offering its own *software*.

With this new *software*, users can take advantage of duplicate print functionality, upgrade to the latest firmware of the printer as soon as it has been published, print and control the printer by connecting it to USB and computer.

# Índex

<b>Resum</b>	<b>1</b>
<b>Resumen</b>	<b>2</b>
<b>Abstract</b>	<b>3</b>
<b>Índex de figures</b>	<b>6</b>
<b>1 Introducció i contextualització</b>	<b>7</b>
1.1 Objectius	9
1.2 Metodologia de treball	9
1.2.1 Què és Scrum?	9
<b>2 Abast</b>	<b>11</b>
2.1 Adaptació del tema	11
2.2 Nova funcionalitat d'impressió duplicada	11
2.3 Connectivitat USB	11
2.4 Gestió d'executables	11
<b>3 Planificació temporal</b>	<b>12</b>
3.1 Planificació estimada	12
3.2 Descripció de les fases del projecte	12
3.2.1 Fase Inicial	12
3.2.2 Primer Sprint	13
3.2.3 Segon Sprint	13
3.2.4 Tercer Sprint	13
3.2.5 Quart Sprint	14
3.2.6 Cinquè Sprint	14
3.2.7 Fase final	14
3.3 Diagrama de Gantt	15
3.4 Pla d'acció i valoració d'alternatives	15
3.5 Recursos	16
<b>4 Gestió econòmica</b>	<b>17</b>
4.1 Identificació i estimació dels costos	17
4.1.1 Recursos humans	17
4.1.2 Recursos hardware	18
4.1.3 Recursos de software	19
4.1.4 Altres recursos	19
4.1.5 Despeses generals	19
4.1.6 Contingències	20
4.1.7 Imprevistos	20
4.1.8 Cost total	21
	4

4.2 Control de gestió	21
<b>5 Sostenibilitat</b>	<b>22</b>
5.1 Autoavaluació sostenibilitat	22
5.2 Matriu de sostenibilitat	22
5.3 Sostenibilitat econòmica	22
5.4 Sostenibilitat ambiental	23
5.5 Sostenibilitat social	23
<b>6 Anàlisi del software Cura</b>	<b>24</b>
6.1 Descripció	24
6.2 Tecnologies	24
6.2.1 Python3	24
6.2.2 QML	24
6.2.3 PyQt	24
6.2.4 OpenGL Shading Language (GLSL)	25
6.2.5 CMake	25
6.3 Estructura	25
6.3.1 CuraEngine	26
6.3.2 LibArcus	26
6.3.3 Uranium	26
6.3.4 Cura	26
6.4 Funcionament	26
<b>7 Desenvolupament de BCN3D Cura</b>	<b>30</b>
7.1 Primers passos	30
7.2 Adaptació del tema	30
7.3 Impressió duplicada	32
7.3.1 Gestió dels elements duplicats	34
7.3.2 Forçar assignació d'extrusors	35
7.3.3 Restaurar el material de l'extrusor	35
7.3.4 Resoldre els paràmetres conflictius	36
7.3.5 Calcular la superfície d'impressió	36
7.3.6 Resultat final	37
7.4 Connectivitat USB	38
7.4.1 Impressió per USB	39
7.4.2 Actualització de firmware	40
7.4.3 Control de la impressora	43
7.5 Gestió dels executables	44
<b>8 Conclusions i treball futur</b>	<b>45</b>
<b>9 Referències</b>	<b>46</b>
9.1 Referències citades	46
9.2 Altres referències	46

# Índex de figures

Diagrama de Gantt	15
Projectes Cura	20
Pantalla selecció d'impressora	27
Pantalla principal	27
Pantalla amb model a la plataforma	28
Pantalla amb model fora la plataforma	28
Barra lateral	29
Icona BCN3D Cura	30
Icona Cura	30
Pantalla d'inici de Cura	31
Pantalla d'inici de BCN3D Cura	31
Pantalla principal Cura	31
Pantalla principal BCN3D Cura	32
Sigmax	33
Desplegable print mode	34
Diagrama de classes SceneNode	34
Duplication	37
Mirror	37
Diagrama de classes connexió USB	38
Pantalla d'actualització de firmware sense impresora connectada	41
Pantalla d'actualització de firmware amb impresora connectada	42
Pantalla procés actualització de firmware	43
Controls USB	44

# 1 Introducció i contextualització

Aquest projecte es du a terme en una empresa (BCN3D<sup>1</sup>) que es dedica al disseny i fabricació d'impressores 3D. És una empresa petita que forma part de la Fundació CIM<sup>2</sup>. Per a entendre algunes de les decisions que es prendran és important tenir en compte que és una empresa amb un pressupost força reduït i conèixer en què consisteix la impressió 3D.

La impressió 3D és l'acció de crear objectes sòlids en tres dimensions a partir d'un arxiu digital. La creació d'un objecte 3D es realitza creant fines capes de material unes sobre les altres successivament fins que es té l'objecte desitjat. Això permet crear objectes amb formes complexes fent servir menys materials que en mètodes tradicionals de fabricació. Per arribar a crear un objecte en 3D no és suficient tenint una impressora. Hi ha una sèrie de passos a realitzar:

## 1. Disseny

Primer de tot necessitem crear el model 3D de l'objecte que volem imprimir. Per a fer-ho hi ha molts programes de disseny que ho permeten, com per exemple SolidWorks o Blender.

## 2. Slicing

Una impressora no és tan intel·ligent com per imprimir un model 3D. Necessita una planificació per saber quins passos ha de seguir. Un cop tenim el model digitalitzat haurem de carregar-lo al *software* encarregat de fer el *slicing*. El *slicing* consisteix en segmentar el model 3D en capes i fer la planificació de totes les accions que haurà de realitzar la impressora per a crear l'objecte. Ens retornarà un fitxer amb totes les instruccions per a la impressora.

## 3. Impressió

Arribats a aquest punt haurem d'enviar el fitxer que ha creat el *software* encarregat de fer el *slicing* a la impressora. Depenent de la impressora això ho podrem fer via wifi, usb, targeta SD, etc.

## 4. Acabat

Segons la tècnica utilitzada per a la impressió l'objecte resultant pot necessitar un procés d'acabat. Com per exemple retirar l'excés de material que hagi pogut quedar a la peça.

Hi ha diferents tipus de tecnologies d'impressió 3D. Tot i que utilitzen tecnologies i processos diferents, els passos descrits anteriorment són comuns per a totes.

En aquest projecte només es treballa amb una tecnologia, Fused Deposition Modeling (FDM)<sup>3</sup>. Aquest és un dels mètodes més utilitzats en el món de la impressió 3D, ja que és bastant més barat que els altres i és accessible per a qualsevol tipus d'usuari.



Els materials usats en aquest mètode són derivats dels plàstics i és un dels punts forts d'aquesta tecnologia, ja que n'hi ha de molts tipus diferents. Materials rígids, flexibles, resistents, de colors i textures diferents, imitació de fusta i de metall, etc.

Aquests materials es troben enrotllats en una bobina en forma de filament. D'aquesta manera podem introduir un extrem al capçal de la impressora i aquest el va estirant a mesura que el va gastant. Al capçal s'escalfa el material perquè es fongui i es fa sortir per un forat molt petit. El capçal es va movent per damunt de la superfície de la impressora dipositant el fil de material que va sortint. Aquesta acció es va fent capa per capa dipositant el material damunt del de la capa anterior fins a obtenir l'objecte final.

#### **Avantatges:**

- Permet fabricar prototips d'un producte de forma ràpida, flexible i econòmica.
- És molt versàtil ja que una impressora 3D no està limitada a produir un tipus de producte, com podria ser el cas d'una cadena de fabricació.
- Pots obtenir els teus productes personalitzats.

#### **Inconvenients:**

- Requereix d'uns mínims coneixements previs per a realitzar impressions de qualitat i dissenys propis.
- Actualment no és assequible per a tothom. Els preus encara són força elevats.

#### **Aplicacions:**

- En medicina s'utilitza per a practicar operacions, com per exemple imprimint una reproducció d'un tumor.
- També en medicina es poden fabricar pròtesis personalitzades.
- Un arquitecte pot imprimir una maqueta del seu disseny.
- Reparació de peces en màquines industrials.

Aquest projecte està enfocat en el *software* encarregat de fer el *slicing*. És un dels punts més importants en la impressió ja que aquest pas té un pes molt important en la qualitat final de l'objecte resultant.

D'aquest projecte se'n podran beneficiar dos parts:

- **Els clients de BCN3D:** Se'ls facilitarà l'ús de les impressores de l'empresa i millorarà la seva experiència d'usuari.
- **Els treballadors de l'empresa:** Els facilitarà la seva feina i reduirà els temps de testeig.
- **La pròpia empresa:** Fer servir un únic *software* en tots els departaments permetrà reduir els errors.

El *slicer*<sup>4</sup> és una aplicació d'escriptori que apart del descrit anteriorment et permet visualitzar els models 3D a la superfície d'impressió i conté una serie de paràmetres que et permeten personalitzar-la. Com per exemple temperatures, velocitats, acceleracions, etc. D'aquests paràmetres dependrà la qualitat de la impressió.

## 1.1 Objectius

Com s'ha explicat a l'apartat anterior, en el procés d'impressió 3D el *slicing* té una gran rellevància. Aquest projecte es centra justament en aquest pas.

Actualment hi ha molts programes que et permeten realitzar aquest pas, s'anomenen *slicers*. Hi ha *slicers* de pagament, gratuïts, *open source*... Alguns són compatibles amb moltes impressores de diferents empreses i altres han estat desenvolupats només pels productes d'una empresa en concret.

Fins ara BCN3D utilitzava una adaptació personalitzada d'un *slicer open source* anomenat *Cura*<sup>5</sup>, però ja fa uns 2 anys va sortir una nova versió d'aquest i és necessari actualitzar-se per oferir als clients totes les possibilitats del mercat actual.

Cal tenir en compte que en aquests últims anys el sector de la impressió 3D ha patit una creixuda important i el *software* que hi està relacionat també. Per tant la necessitat d'actualitzar-se és gran i urgent.

L'objectiu d'aquest projecte és actualitzar-se a l'última versió de *Cura* i adaptar-lo per a satisfer les necessitats dels productes de BCN3D.

Fent això es vol aconseguir:

- **Reforçar la marca BCN3D:**

Oferir un *software* propi que es diferencia de la resta.

- **Facilitar al client l'ús dels productes i millorar l'experiència d'usuari:**

Els *slicers* genèrics per a moltes impressores poden tenir funcionalitats no aptes per a algunes i intruccions que poden confondre els usuaris.

- **Aprofitar totes les característiques particulars de les impressores de BCN3D:**

Cada impressora té unes propietats específiques i unes particularitats, per tant tenint un *software* propi les pots expressar al màxim sense limitacions.

## 1.2 Metodologia de treball

Aquest projecte es realitza amb un equip de dues persones. En el meu cas tinc el rol de desenvolupador de *software*, que sóc l'encarregat de desenvolupar les noves funcionalitats i adaptacions del *software*.

L'altre persona té coneixements avançats d'impressió 3D i és l'encarregada de perfeccionar els paràmetres d'impressió per a aconseguir la millor qualitat possible.

Per a organitzar els projectes de l'empresa utilitzem *Asana*<sup>6</sup>. Una eina per a la gestió de projectes en equip.

L'equip de desenvolupament del projecte s'organitza mitjançant *Scrum*<sup>7</sup>.

### 1.2.1 Què és *Scrum*?

*Scrum* és un marc de treball per a la gestió de projectes en equip. Per la forma de treballar ens permet reaccionar als canvis d'una manera àgil.

El primer que s'ha de fer es definir els objectius del projecte i definir totes les tasques que s'hauran de fer per a complir-los. Després s'ha d'estimar l'esforç que requereix el desenvolupament de cada tasca i ordenar-les per prioritat.

Totes aquestes tasques formen el que s'anomena *product backlog*.

Arribats aquest punt es comença la planificació dels *sprints*. Els *sprints* en *Scrum* són uns períodes de durada fixa, es recomana que sigui del voltant de dues setmanes, en els que l'equip es marcarà l'objectiu de realitzar una sèrie de tasques. D'aquesta manera es pot ser àgil als canvis que van sortint durant el projecte.

Al finalitzar cada *sprint* la idea és que hi hagi un producte funcional que aportï valor al client. Aleshores es planifica el següent *sprint* i així successivament.

S'ha escollit realitzar el projecte sota aquest marc de treball perquè ens ajuda a poder treure versions funcionals del producte i anar obtenint *feedback* del usuaris finals, ja que aquest és un dels objectius més importants.

La idea és que al final de cada *sprint* es pugui treure una nova versió del *software* que incrementi el seu valor.

Com a control de version utilitzem *git* i el codi està allotjat a *Github*<sup>8</sup>.

Els errors i suggeriments que ens vulguin traslladar els usuaris ens ho podran fer saber per *Github*. Això ens permet validar si s'està anant pel bon camí.

## 2 Abast

Aquest projecte es basa en adaptar el *slicer Cura*, que és una aplicació d'escriptori, per a complir les necessitats de BCN3D.

Això implica el desenvolupament o adaptació de les següents funcionalitats:

### 2.1 Adaptació del tema

Donar un aspecte diferenciat al *software* usant els colors i les fonts pròpies de BCN3D, així com introduir el logo de l'empresa. Sempre fent referència a que és una versió adaptada del *software open source* desenvolupat per *Ultimaker*<sup>9</sup> juntament amb la comunitat.

### 2.2 Nova funcionalitat d'impressió duplicada

Una de les característiques de la nova impressora de BCN3D és que té dos capçals i que pot imprimir dos peces iguals al mateix temps. El problema és que aquesta funcionalitat no existeix a *Cura*, per tant s'ha d'implementar des de 0. Això implica per una part l'aspecte visual, és a dir, que es mostrin totes les peces de la superfície duplicades. Per altre banda totes les funcionalitats han de seguir funcionant, ja que és una funcionalitat que afecta a molts dels paràmetres d'impressió i al comportament d'altres funcionalitats del programa. Per l'últim s'haurà de garantir que les instruccions que s'enviaran a la impressora seran les correctes.

### 2.3 Connectivitat USB

Aquesta és una funcionalitat que ja està implementada, però de forma genèrica per a moltes impressores. En el cas de les impressores de BCN3D, això implica que no tingui un funcionament correcte ni fiable.

Millorant aquesta connectivitat es vol:

- **Imprimir per USB:** Això permet una millor monitorització de la impressió.
- **Actualitzar el firmware de la impressora:** Es vol poder actualitzar-lo de forma automàtica descarregant-se l'última versió d'internet o carregant-ne un des del propi ordinador.
- **Control de la impressora:** Poder enviar instruccions a la impressora. Com per exemple: moure capçals, escalfar la superfícies d'impressió, etc.

### 2.4 Gestió d'executables

L'aplicació ha d'estar disponible per a linux, windows i mac. Periòdicament es vol anar traient noves versions per a que les puguin utilitzar els clients.

## 3 Planificació temporal

### 3.1 Planificació estimada

Aquest projecte s'ha realitzat en el període comprès entre el 16/09/2017 i el 16/02/2018. La durada ha estat de 5 mesos i les hores dedicades han estat 735 aproximadament. Ja que és un treball de modalitat B.

### 3.2 Descripció de les fases del projecte

Aquest projecte s'ha desenvolupat utilitzant *Scrum*, que és una metodologia àgil. Per tant constarà d'una fase inicial en la que es prepararà tot l'entorn per a poder començar amb el desenvolupament del projecte i després s'aniran fent iteracions afegint poc a poc valor al producte.

En la metodologia *Scrum* aquestes iteracions s'anomenen *sprints*. El cap d'enginyeria de l'empresa, que també és el director del projecte, és l'encarregat de prioritzar les tasques i decidir quines es faran a cada *sprint*. En aquest projecte s'ha definit que cada *sprint* tindrà una duració de dues setmanes. Aleshores cada dues setmanes es farà una reunió per a validar els resultats del *sprint* acabat i per a planificar el següent.

Al final de cada *sprint* es farà una demostració del resultat a treballadors de diferents àrees de l'empresa. Entre tots es decidirà si les noves funcionalitats implementades es donen per bones i tothom tindrà la possibilitat d'aportar noves idees. Si es considera oportú es treurà una versió beta del producte per a poder anar rebent informació dels usuaris que l'han fet servir.

En cas de que en un *sprint* es tregui una versió beta, al finalitzar el següent *sprint*, es tindrà en compte les aportacions rebudes dels usuaris que han fet servir el producte per a la planificació del pròxim *sprint*. Ja que és probable que hi hagin errors que haguem passat per alt o que ens adonem que estem tirant pel camí equivocat.

#### 3.2.1 Fase Inicial

Com que aquest projecte consisteix en l'adaptació d'un *software open source* ja existent hi haurà un procés de familiarització amb les tecnologies que utilitza aquest *software*.

Serà la fase més llarga del projecte ja que no es tenen coneixements previs del producte a desenvolupar. Un cop endinsats dins el nou *software* s'estudiarà quines de les funcionalitats que es volen afegir o adaptar són factibles d'implementar.

S'haurà de descobrir com compilar-lo i executar-lo i les eines que calen per a fer-ho. Després s'haurà de veure com crear els executables per a Linux, Mac i Windows. Ja que es vol oferir el producte a usuaris finals.

Aquesta fase es calcula que durarà un mes i mig aproximadament. Quan es doni per finalitzada perquè es considera que s'han assolit els seus objectius es començarà amb els *sprints*.

### 3.2.2 Primer *Sprint*

Al començar el primer *sprint* ja tindrem l'entorn de desenvolupament creat i estarem una mica familiaritzats amb l'estructura del *software*.

L'objectiu d'aquest *sprint* serà tenir tres executables del programa (Linux, Mac i Windows) amb les característiques de BCN3D. És a dir que el producte no crei conflictes amb Ultimaker, que és l'empresa que desenvolupa el producte juntament amb la comunitat. Per exemple: per reportar errors no enllaçar la web d'Ultimaker, no enviar informació a Ultimaker. També voldrem que en els llocs on calgui aparegui el nou nom: BCN3D Cura. Per últim només es podran escollir les impressores de l'empresa amb els materials corresponents.

### 3.2.3 Segon *Sprint*

En aquest *sprint* ja es començarà a desenvolupar noves funcionalitats. Concretament el que es farà és desenvolupar el mode d'impressió per duplicat i en mirall. És la funcionalitat més important que tindrà l'adaptació de BCN3D i és un del motius més importants per als que té sentit el projecte.

L'objectiu d'aquest *sprint* és poder visualitzar els models 3D per duplicat i en mirall i poder fer les operacions bàsiques sobre aquests models 3D: translació, escalat i mirall.

### 3.2.4 Tercer *Sprint*

En aquest *sprint* tindrem la funcionalitat dels modes d'impressió desenvolupada en un punt inicial, però sense tenir en compte totes les implicacions que pot tenir en altres funcionalitats.

L'objectiu d'aquest *sprint* és desenvolupar les operacions complexes sobre els models 3D i acabar de polir les implicacions d'aquesta funcionalitat sobre la resta, de manera que el comportament és l'esperat en tots els casos.

### 3.2.5 Quart *Sprint*

En aquest *sprint* ja tenim un producte propi amb una funcionalitat que el diferencia.

L'objectiu d'aquest *sprint* serà donar un toc personalitzat al producte adaptant el tema per a que s'identifiqui amb BCN3D. També farem els canvis necessaris per a que les impressores de BCN3D es detectin correctament per USB.

### 3.2.6 Cinquè *Sprint*

Aquest és l'últim *sprint*.

L'objectiu d'aquest *sprint* serà desenvolupar l'actualització del *firmware* per USB de les impressores de BCN3D. També s'haurà de poder imprimir a través de USB.

### 3.2.7 Fase final

Aquesta fase tindrà una durada d'un mes, ja que seria normal que durant el desenvolupament surtin imprevistos que necessitin ser corregits.

També s'hauran de corregir els errors que hagin trobat els usuaris que hauran fet servir les versions betes del producte.

### 3.3 Diagrama de Gantt



Diagrama de Gantt

### 3.4 Pla d'acció i valoració d'alternatives

La fase final del projecte té una duració llarga ja que es té en compte que es pot donar el cas que les tasques planificades per a cada *sprint* no es compleixin. Ja sigui perquè s'han estimat equivocadament o perquè ha sorgit algun imprevist i no se li ha pogut dedicar el temps previst.

Les metodologies àgils ens permeten adaptar-nos als canvis, gràcies a la durada curta de les iteracions. Per això en cas de que sigui necessari podrem donar prioritat a les tasques més crítiques per a aconseguir els objectius del projecte dins del temps establert.



## 3.5 Recursos

Per al desenvolupament del projecte s'han utilitzat els materials i eines següents:

Recurs	Tipus	Finalitat
Ordinador HP (Linux, Windows)	Eina de desenvolupament	Desenvolupar i crear executables.
Mac mini	Eina de desenvolupament	Crear executable per Mac.
Git	Eina de desenvolupament	Control de versions.
PyCharm	Eina de desenvolupament	IDE per a programar el <i>software</i> .
GitHub	Eina de desenvolupament	Emmagatzemar el codi font i control de
Google Drive	Eina de gestió	Documents estadístics sobre el desenvolupament del projecte.
Asana	Eina de comunicació	Gestió de les tasques a realitzar.
Impresora 3D Sigma	Eina de testeig	Testeig del producte.
Impresora 3D Sigmax	Eina de testeig	Testeig del producte.
Material d'impressió	Eina de testeig	Testeig del producte.
Google Chrome	Eina de recerca	Recerca de informació.
Google Hangouts	Eina de comunicació	Comunicació entre els membres de l'empresa.

## 4 Gestió econòmica

### 4.1 Identificació i estimació dels costos

#### 4.1.1 Recursos humans

Per a calcular quins seran els costos relacionats amb els recursos humans ens basarem en les taules retributives de BCN3D.

En aquest projecte hi intervenen tres persones: una amb el rol de cap de projecte i dues amb el rol de desenvolupadors. El cap de projecte i un desenvolupador tindran el sou corresponent segons la política retributiva de l'empresa. L'altre desenvolupador tindrà el sou que marca el conveni de pràctiques de l'UPC.

En el cas del conveni de pràctiques de l'UPC el cost per hora està especificat. Per a calcular el cost per hora dels altres dos treballadors farem el càlcul següent:

$$\text{€/hora} = (\text{Sou anual} / 14 \text{ pagues}) / (40\text{h} * 4 \text{ setmanes})$$

Rol	€/hora
Cap de projecte	12,15 €
Desenvolupador BCN3D	8,83 €
Desenvolupador UPC	8 €

Un cop tenim el cost per hora de cada treballador del projecte, podrem estimar el cost total d'aquest. Per a fer-ho ens basarem en l'estimació temporal de totes les tasques a realitzar durant el projecte.

Tasca	Hores	Recurs	Cost (€)
<b>Fase inicial</b>	<b>240</b>		<b>1080</b>
Preparació de l'entorn	80	Desenvolupador UPC	360

Generació d'executables	80	Desenvolupador UPC	360
Aprenentatge autònom	80	Desenvolupador UPC	360
<b>Sprint 1-5</b>	<b>80</b>		<b>1254.2</b>
Planificació	8	Cap del projecte	97.2
Desenvolupament	68	Desenvolupadors x2	1108.4
Demostració	4	Cap del projecte	48.6
<b>Fase final</b>	<b>160</b>		<b>1944</b>
Correcció d'errors	80	Desenvolupadors x2	1304
Documentació TFG	80	Desenvolupador UPC	640
<b>Total</b>			<b>1080 + (1254.2*5) + 1944 = 9295</b>

#### 4.1.2 Recursos hardware

Per a calcular el cost d'amortització dels recursos de hardware suposarem que aquests tindran una vida útil de 3 anys. Per tant utilitzarem el càlcul següent:

$\text{Cos d'amortització} = \text{Preu(€)} / (3 \text{ anys} * (8\text{h/dia} * 249 \text{ dies hàbils any}))$

Recurs	Preu (€)	Utilització (h)	Cost d'amortització (€/h)	Cost estimat (€)
Mac mini	555	50	0.09	4.5
Ordinador HP	800	600	0.13	78
Ordinador HP	800	600	0.13	78
Ordinador HP	800	80	0.13	10.4
Impressora Sigma	900	200	0.15	30
Impressora Sigmax	1100	200	0.18	36
<b>Total</b>				<b>236.9</b>

#### 4.1.3 Recursos de *software*

Per a calcular el cost d'amortització dels recursos de *software* suposarem que aquests tindran una llicència de 3 anys. Per tant utilitzarem el càlcul següent:

$\text{Cos d'amortització} = \text{Preu(€)} / (3 \text{ anys} * (8\text{h/dia} * 249 \text{ dies hàbils any}))$

Recurs	Preu (€)	Utilització (h)	Cost d'amortització (€/h)	Cost estimat (€)
Windows 7	75	600	0.01	6
Google Drive	0		0	0
GitHub	0		0	0
Google Hangouts	0		0	0
Asana	0		0	0
PyCharm	0		0	0
<b>Total</b>				<b>6</b>

#### 4.1.4 Altres recursos

Per a poder validar moltes de les funcionalitats d'aquest projecte és necessari posar-les a prova realitzant impressions 3D. Per tant és important tenir en compte el cost del material per a imprimir.

Una bobina de material té un cost per a l'empresa de 12€ i es calcula que cada setmana es gastaran unes dos bobines de material. Per tant tenint en compte que el projecte durarà unes 20 setmanes el cost estimat de material serà el següent:  $12 * 2 * 20 = 480\text{€}$ .

#### 4.1.5 Despeses generals

Com a despeses generals tindrem en compte el cost del lloguer de les oficines, però només una part ja que aquestes les usen també molts altres treballadors. El càlcul seria el següent:

$300\text{€/mes} * 5 \text{ mesos} = 1500\text{€}$

#### 4.1.6 Contingències

Fixarem com a nivell de contingència un 15%. Aquest 15% del cost total del projecte es reservarà per a possibles imprevistos que apareguin.

	Percentatge	Preu (€)	Cost (€)
<b>Cost directe</b>	15 %	9537.9	1430.68
<b>Cost indirecte</b>	15 %	1500	225
<b>Total</b>			<b>1655.68</b>

#### 4.1.7 Imprevistos

Els imprevistos que es poden donar serien que alguna impressora o algun ordinador es pogués espatllar. Estimem que la probabilitat de que s'espatlli una impressora és d'un 15% i la de que s'espatlli un ordinador és d'un 10%. Per a calcular el cost de la impressora agafarem el preu mig entre els dos models (Sigma i Sigmax).

Imprevist	Probabilitat	Quantitat	Cost (€)
Avaria ordinador	10%	1	800
Avaria impressora	15%	1	1000
<b>Total</b>			<b>1800</b>

#### 4.1.8 Cost total

Concepte	Cost (€)
Recursos humans	9295
Recursos hardware	236.9
Recursos <i>software</i>	6
Altres recursos	480
Despeses generals	1500
Contingències	1655.68
Imprevistos	1800
<b>Total</b>	<b>14973.58</b>

## 4.2 Control de gestió

Considerem que els costos dels materials d'impressió són els que poden tenir una desviació més gran, ja que no és fàcil estimar la quantitat que es gastarà. Per això es realitzarà un seguiment del que es gasta per a veure la desviació respecte l'estimat.

Respecte els costos de recursos humans, considerem que la desviació es pot produir en les hores dedicades pel cap del projecte. Ja que és qui realitza una aportació a temps parcial i l'estimació pot haver estat errònia. Es realitzarà un control de les hores que dedica al projecte.

Al final tindrem les dades necessàries per a veure si les estimacions s'han complert o quina desviació hi ha hagut.

## 5 Sostenibilitat

### 5.1 Autoavaluació sostenibilitat

Després de realitzar l'enquesta sobre sostenibilitat m'he adonat de que en tinc pocs coneixements i de que alguns aspectes ni te'ls plantejes a l'hora de realitzar un projecte. Per altre banda també me n'he adonat que alguns aspectes ja els tenia en compte sense ser-ne del tot conscient.

Els punts en els que crec que necessito millorar són els següents:

- Saber medir l'impacte de les tecnologies TIC amb els indicadors adequats.
- Conèixer mètodes per a minimitzar l'impacte ambiental.
- Coneixements sobre economia social.
- Coneixements del principis deontològics.

En canvi alguns dels aspectes que ja tenia en compte són els següents:

- Impacte dels projectes TIC sobre el medi ambient.
- Com ajudarà a la societat un projecte TIC.
- Reutilització de recursos.

Aquest projecte m'ajudarà a consolidar els punts forts i a aprendre aquells aspectes que fins ara ni em plantejava a l'hora de realitzar un projecte.

### 5.2 Matriu de sostenibilitat

Sostenibilitat	Ambiental	Econòmica	Social
PPP	Anàlisi de l'impacte ambiental	Anàlisi cost del projecte	Aportació a nivell personal
Valoració	6	8	8

### 5.3 Sostenibilitat econòmica

El *software* desenvolupat en aquest projecte no generarà un benefici econòmic directe, ja que es podrà descarregar de forma gratuïta. L'objectiu és el de oferir un millor servei als clients i potenciar el producte actual de l'empresa.

Per a poder realitzar el producte amb el mínim cost possible s'ha intentat prioritzar bé les tasques a realitzar de manera que ens assegurem de fer les que aportin més valor. D'aquesta manera disminueix el risc que té per a l'empresa el desenvolupament d'un producte d'aquestes característiques.

## 5.4 Sostenibilitat ambiental

L'impacte ambiental que tindrà el projecte serà el de l'energia consumida, els recursos hardware (que en un futur seran residus) i els materials d'impressió. En el cas de l'energia consumida i els recursos hardware no hi ha gaire marge de millora. Però en el cas dels materials d'impressió el material més usat serà el PLA, que és un material biodegradable. D'aquesta manera es minimitzarà l'impacte ambiental.

## 5.5 Sostenibilitat social

Aquest projecte m'aportarà l'oportunitat de conèixer un món com el de la impressió 3D i l'oportunitat de desenvolupar un projecte on he d'aprendre i tenir en compte altres aspectes que no estan relacionats amb la informàtica.

El desenvolupament d'aquest projecte representarà una millora per als clients de l'empresa que tinguin els seus productes, ja que els facilitarà l'utilització d'aquests i permetrà expressar totes les seves possibilitats al màxim. Precisament això és una necessitat bàsica de l'empresa ja que s'estava quedant enrere respecte la competència.



## 6 Anàlisi del *software Cura*

### 6.1 Descripció

*Cura* és un *software open source* desenvolupat per *Ultimaker* juntament amb la comunitat. La seva funció és la de preparar models per a poder imprimir-los amb una impressora 3D.

Permet carregar els models digitalitzats en 3D i visualitzar-los sobre una plataforma que representa l'àrea d'impressió de la impressora. Un cop tenim els models sobre la plataforma se'ls pot canviar les dimensions, la posició en la que s'imprimiran, es poden rotar i agrupar. També es poden canviar una sèrie de paràmetres que influiran en el resultat final del *slicing* i per tant de la impressió. Per exemple es pot definir a quina temperatura s'haurà d'imprimir el material.

### 6.2 Tecnologies

#### 6.2.1 Python3

Llenguatge en el que *Cura* està desenvolupat. És un llenguatge de programació d'alt nivell desenvolupat com a projecte *open source* i és administrat per l'empresa *Python Software Foundation*. Suporta diversos paradigmes de programació, com programació orientada a objectes, que és com *Cura* està desenvolupat. La llegibilitat i simplicitat de Python ajuda a poder expressar els mateixos conceptes amb menys línies de codi i de manera més entenedora.

#### 6.2.2 QML

És el llenguatge de programació amb el que està desenvolupada la interfície d'usuari de *Cura*. QML té una sintaxi llegible, declarativa i amb suport per a expressions JavaScript.

#### 6.2.3 PyQt

És un *binding* de la biblioteca gràfica Qt per a Python. Permet treure profit de tots els avantatges de Qt, però amb la simplicitat d'un llenguatge com Python.

### 6.2.4 OpenGL Shading Language (GLSL)

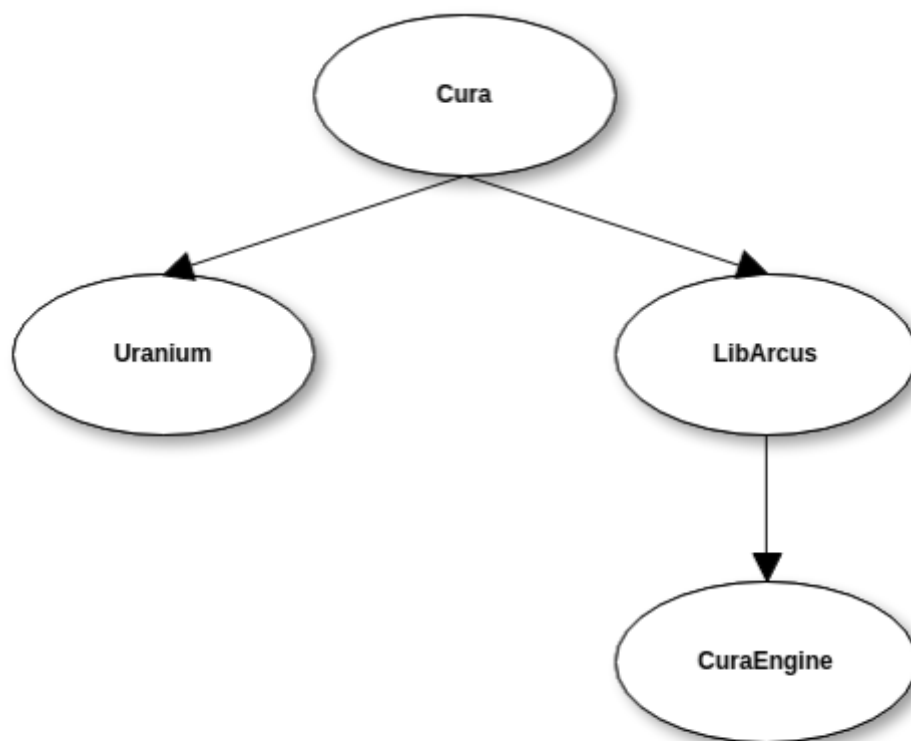
És un llenguatge d'alt nivell d'ombrejat amb una sintaxi basada en el llenguatge de programació C.

### 6.2.5 CMake

És una família d'eines *open source* i multiplataforma dissenyada per a construir, testear i empaquetar *software*. S'utilitza per a controlar la el procés de compilació del *software* fent servir una plataforma senzilla i uns fitxers de configuració independents al compilador.

## 6.3 Estructura

*Cura* està format per diversos projectes:



Projectes Cura

### 6.3.1 CuraEngine

És una aplicació de consola C++ per a la generació d'arxius GCode a partir d'un model 3D digitalitzat. És el que utilitza *Cura* per a fer el *slicing*.

Aquest projecte no ha estat modificat per a la realització del TFG.

### 6.3.2 LibArcus

És una llibreria que conté C++ i *bindings* de Python per a crear un *socket* en un *thread* per enviar i rebre missatges. Aquesta llibreria ha estat dissenyada per a facilitar la comunicació entre *Cura* i el *CuraEngine*.

Aquest projecte no ha estat modificat per a la realització del TFG.

### 6.3.3 Uranium

És un framework desenvolupat per *Ultimaker* principalment per al desenvolupament de *Cura*. Les classes de Python i objectes de QML que proporciona faciliten i simplifiquen el desenvolupament de *Cura*.

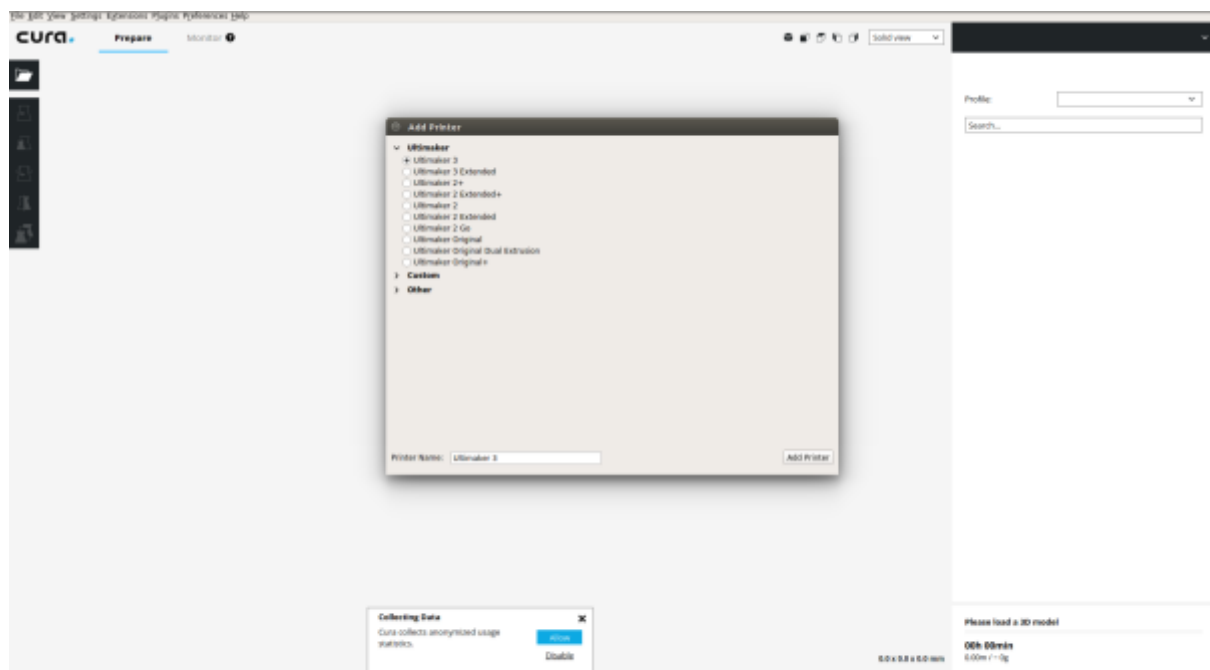
Per a desenvolupar algunes de les funcionalitats definides a l'abast s'ha hagut de modificar aquest projecte.

### 6.3.4 Cura

És el projecte principal que conté l'aplicació d'escriptori. Permet utilitzar el *CuraEngine* per a fer el *slicing* de models 3D proporcionant una interfície gràfica. Amb aquesta interfície es poden editar una sèrie de paràmetres que es tindran en compte a l'hora de fer el *slicing*. També permet visualitzar i moure el model 3D sobre la plataforma de la impressora per indicar on s'haurà d'imprimir.

## 6.4 Funcionament

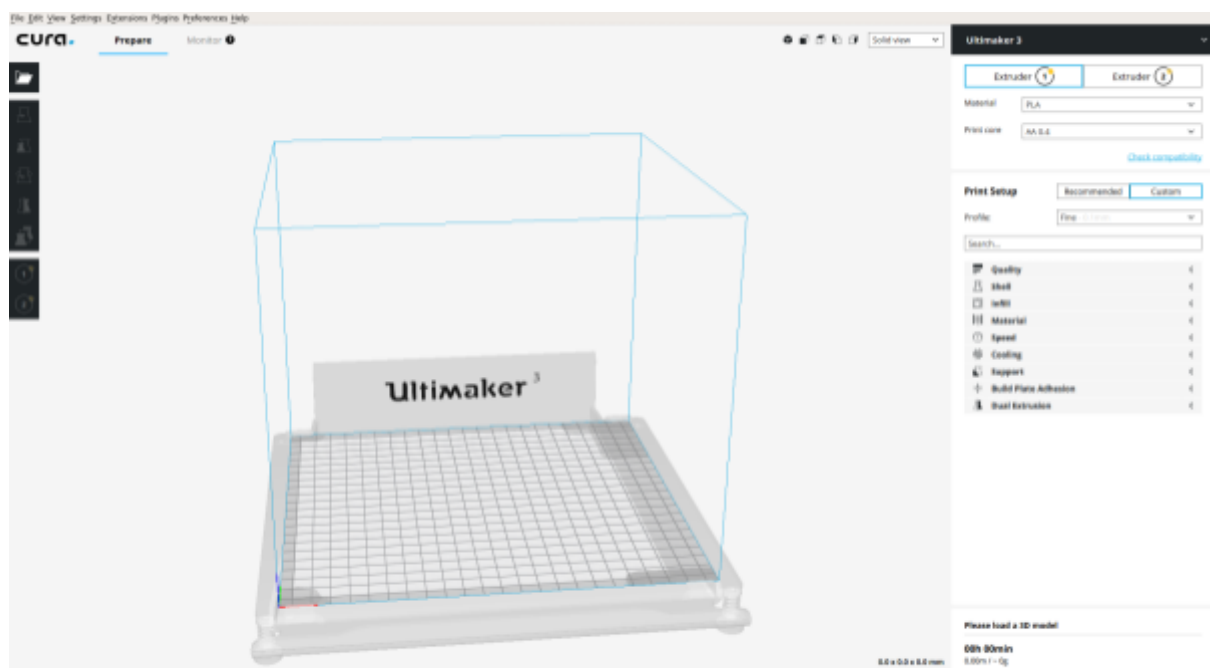
Al arrancar Cura per primer cop ens apareixerà una pantalla per seleccionar quin és el model de la nostra impressora. Les impressores de BCN3D no apareixen en el Cura d'*Ultimaker*, per tant seleccionarem una de les seves impressores per a fer-la servir d'exemple.



Pantalla selecció d'impressora

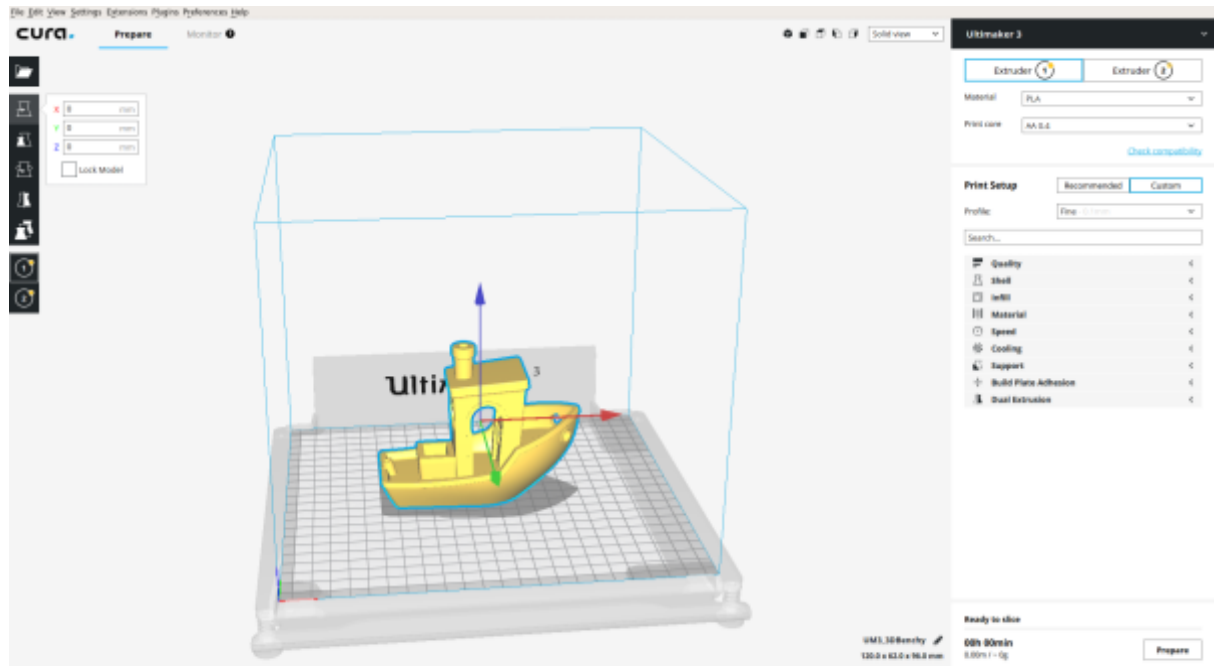
Un cop seleccionat el model d'impressora que farem servir veurem al centre de la pantalla una plataforma amb un cub blau que representa el volum d'impressió. Les parts ombrejades de la plataforma indiquen que en aquella zona no es pot imprimir.

Per a carregar un model 3D digitalitzat s'ha de fer clic al botó amb la icona d'una carpeta que hi ha a la columna de l'esquerra i seleccionar un fitxer 'stl'.



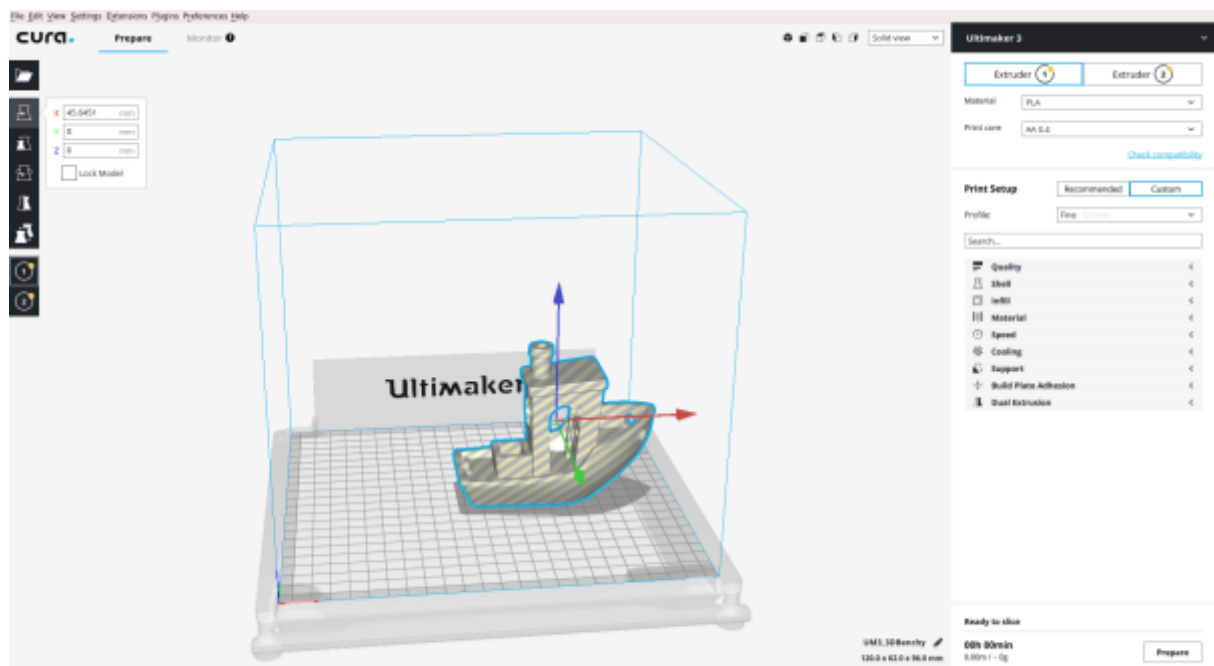
Pantalla principal

Un cop tenim el model sobre la plataforma el podem moure, escalar, rotar, multiplicar...



Pantalla amb model a la plataforma

En cas que el model surti de la part de la plataforma on es pot imprimir es mostrarà com a la següent imatge:



Pantalla amb model fora la plataforma

Un cop tenim el model col·locat dins la plataforma d'impressió ja podem fer el *slice* per a obtenir l'arxiu GCode, que és el que posarem a la impressora per a imprimir el model. Per a fer el *slice* només cal apretar el botó que hi ha abaix a la dreta i quan acabi podrem guardar l'arxiu resultant.

En la barra lateral apareixen tots els components necessaris per a poder definir com es durà a terme la impressió.

A dalt de tot podem veure el nom del model d'impressora que estem utilitzant. Clicant-hi podem canviar de model. Just a sota veiem que hi ha 2 extrusors, això vol dir que la màquina té dos punts de sortida de material i podem seleccionar el material que sortirà per cada punt.

Més avall veiem una sèrie de paràmetres classificats per categories. En la imatge algunes categories es mostren desplegadas i d'altres no. Segons el material escollit aquests paràmetres tindran uns valors o uns altres. A més l'usuari pot modificar el valor d'aquests un per un. És molt important que els paràmetres tinguin un valor òptim per al model que es vol imprimir, ja que aquests valor són els que s'utilitzaran a l'hora de fer el *slicing* i determinaran la qualitat del resultat final.

Per acabar només cal fer clic al botó *Prepare* per tal de fer el *slicing* i poder guardar l'arxiu GCode resultant.

The image shows the 'Print Setup' menu in the Ultimaker 3 software. At the top, there's a header 'Ultimaker 3' with a dropdown arrow. Below it, two tabs are visible: 'Extruder 1' (selected) and 'Extruder 2'. Under 'Extruder 1', there are dropdowns for 'Material' (set to 'PLA') and 'Print core' (set to 'AA 0.4'). A link 'Check compatibility' is present. The 'Print Setup' section has two tabs: 'Recommended' and 'Custom' (selected). Below this is a 'Profile' dropdown set to 'Fine - 0.1mm' and a 'Search...' input field. The settings are organized into expandable categories: 'Quality' (Layer Height: 0.1 mm), 'Shell', 'Infill', 'Material' (Printing Temperature: 200 °C, Build Plate Temperature: 60 °C, Diameter: 2.85 mm, Flow: 100 %, Enable Retraction: checked), 'Speed', 'Cooling', 'Support', and 'Build Plate Adhesion' (Build Plate Adhesion Type: Brim, Build Plate Adhesion Extruder: Extruder 1, Brim Width: 7 mm). At the bottom, there's a 'Dual Extrusion' category. The bottom of the interface shows 'Ready to slice', a timer '00h 00min' with '0.00m / ~ 0g', and a 'Prepare' button.

Barra lateral

## 7 Desenvolupament de BCN3D Cura

### 7.1 Primers passos

Per a poder començar a desenvolupar el *software* el primer que necessitàvem era compilar i executar el *software* original desenvolupat per Ultimaker. En els repositoris de Github hi ha les instruccions necessàries per a compilar els diferents projectes dels que Cura depèn.

Al principi vam compilar tots aquests projectes un per un, però més endavant vam descobrir que Ultimaker tenia uns scripts per a compilar totes les dependències necessàries i per a generar els executables de Cura.

Aquest procés ha estat un dels més crítics de tot el projecte, ja que era necessari superar-lo per a poder seguir amb el desenvolupament del projecte. Varen anar sorgint molts problemes de compilació, que en alguns casos no va ser fàcil trobar-ne una solució.

### 7.2 Adaptació del tema

Un dels motius per als que es volia tenir un *software* propi era per a donar més valor a la marca BCN3D. Per això s'ha volgut donar un aspecte personalitzar diferent del *Cura* desenvolupat per *ultimaker*.

Primer de tot el que s'ha fet és canviar el nom i passar-se a dir *BCN3D Cura*. També s'ha modificat l'ícona que es mostra quan el programa està obert.



Icona BCN3D Cura



Icona Cura

Al iniciar *Cura* apareix una pantalla mentre s'està carregant tot el que és necessari. Aquesta pantalla també s'ha modificat afegint el logo de BCN3D i modificant una mica l'estètica i la posició dels diferents components.

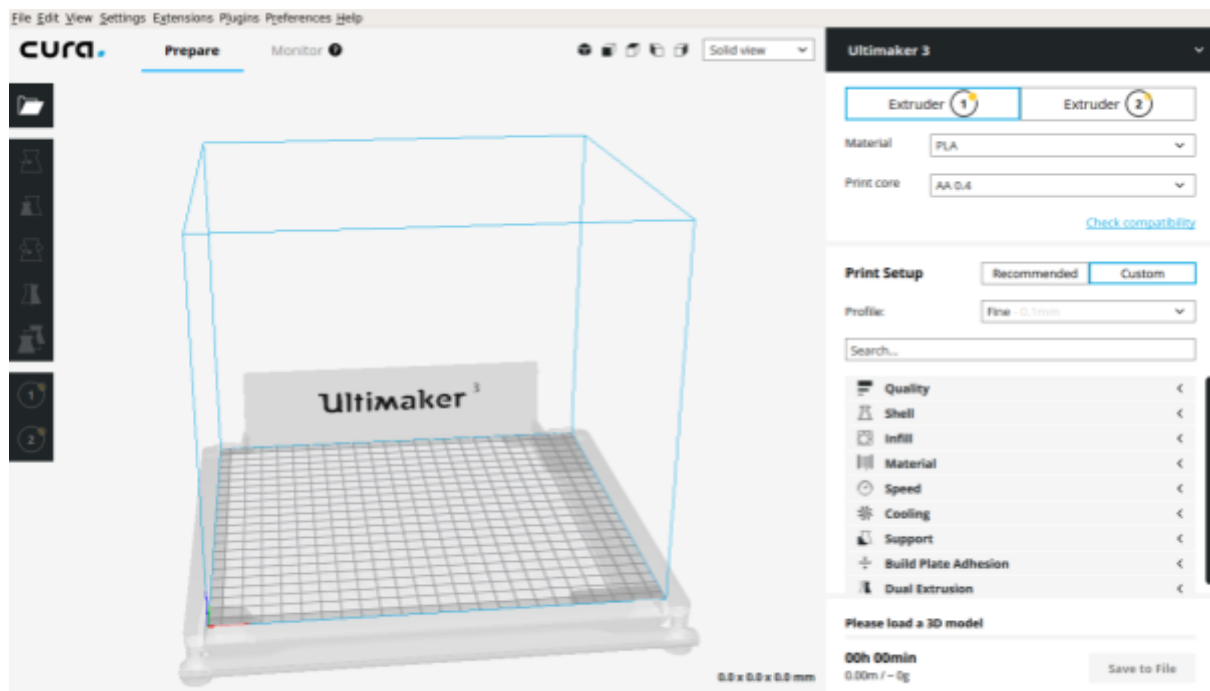


Pantalla d'inici de Cura



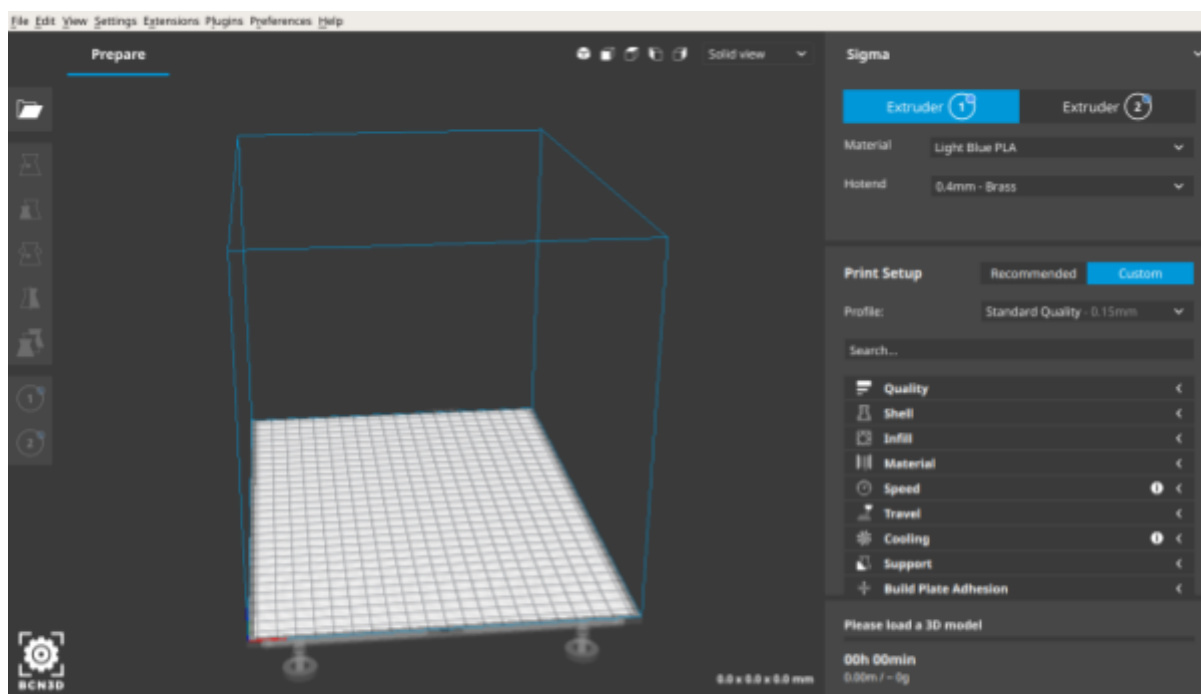
Pantalla d'inici de BCN3D Cura

L'última modificació que s'ha fet sobre l'aparença són els colors de la interfície gràfica. L'equip de disseny va fer una proposta amb els colors que millor definien la marca BCN3D i aquests són els que predominen en el *software* BCN3D Cura.



Pantalla principal Cura





Pantalla principal BCN3D Cura

Com podem veure a les imatges s'ha canviat el logo de *Cura* que hi ha a dalt a l'esquerra pel logo de BNC3D que ha passat a estar abaix a l'esquerra. El canvi més imporant són els colors, on ara predominen un grisos foscots amb tocs blaus quan es cliquen els botons o si passa per sobre.

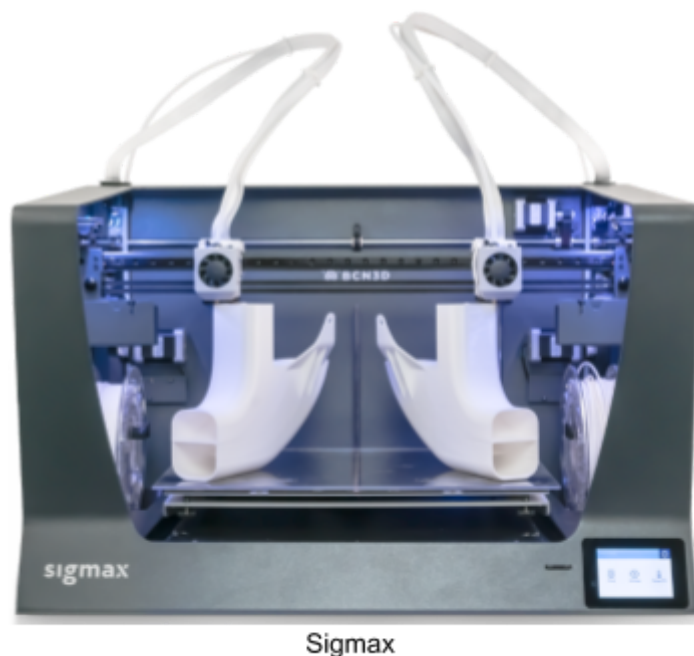
Per a poder realitzar totes aquestes modificacions, *Cura* té un fitxer "json" on estan definits tots els colors i tamany dels diferents components de la interfície gràfica, per tant per fer canvis senzills sobre aquesta només cal editar aquest fitxer "json".

## 7.3 Impressió duplicada

La funcionalitat d'impressió duplicada és una de les novetats que havia de tenir el nou *software* ja que és una funcionalitat específica de les impressores de BCN3D.

A la imatge podem veure una de les impressores de BCN3D, la *sigmax*. Aquesta impressora té dos capçals independents i per tant pot imprimir dues peces iguals.

Com podem veure a la imatge la impressió està feta en mirall, és a dir els dos capçals fan els mateixos moviments en l'eix Y, però invertits en l'eix X. En el mode de duplicació els moviments són exactament els mateixos en tots els eixos.



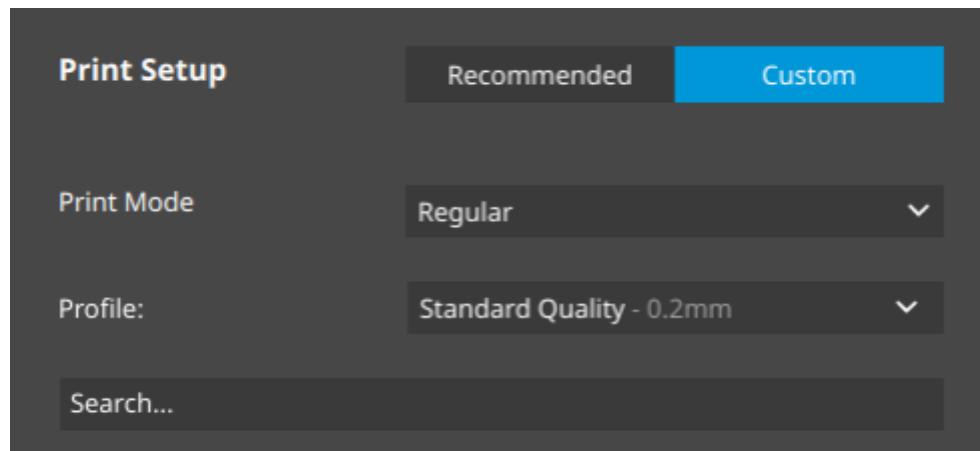
SigmaX

Com s'ha vist en l'apartat anterior al carregar un model sobre la plataforma d'impressió, aquest es pot moure per tota la superfície, però només hi ha un model. El que volem és crear uns nous modes d'impressió que quan estiguin seleccionats la superfície d'impressió quedi reduïda a la meitat i tots els models que hi hagi a la part esquerra es mostrin duplicats a la part dreta.

Primer de tot el que necessitem és un selector que ens permeti canviar el mode d'impressió. Tindrem tres modes d'impressió:

- **Regular:** És el mode d'impressió normal, no és necessari fer cap modificació respecte el *Cura* original.
- **Duplication:** És el mode d'impressió per duplicat. Tots els models es veuran duplicats a la part dreta de la plataforma d'impressió.
- **Mirror:** És el mode d'impressió en mirall. Tots els models es veuran a la part dreta de la plataforma d'impressió amb les coordenades de l'eix X invertides.

Aquest selector amb els tres modes d'impressió s'ha col·locat al lateral dret juntament amb tots els paràmetres d'impressió. Per a fer-ho hem creat un component QML que és un desplegable amb els tres modes d'impressió i que només és visible per als models que el suporten.



Desplegable Print Mode

Per a gestionar tots els aspectes que s'han de tenir en compte a l'hora de canviar de mode d'impressió hem creat la classe de python `PrintModeManager`. Aquesta classe s'ha creat fent servir el patró de disseny *singleton*, és a dir, només hi pot haver una sola instància. En el constructor comprova si ja hi ha una instància existent i en cas de que la hi hagi retorna un missatge d'error, si no n'hi ha cap el que fa és crear-la.

La classe `PrintModeManager` realitza les següents funcions:

### 7.3.1 Gestió dels elements duplicats

Tots els elements que formen part de l'escena són del tipus `SceneNode`. Com es pot veure en el diagrama hi ha diferents tipus de `SceneNode`.

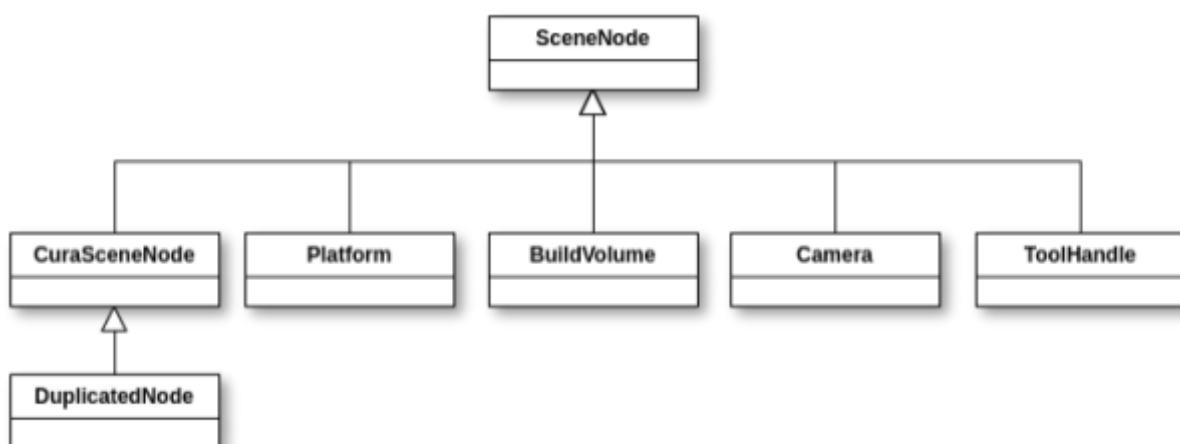


Diagrama de classes `SceneNode`

- **CuraSceneNode:** Representa els models que s'han carregat en forma d'arxiu "stl". Guarda informació geomètrica sobre aquest: BoundingBox, posició en la plataforma, rotació, tamany...
- **DuplicatedNode:** És subclasse de CuraSceneNode. Representa els models duplicats i només es mostraran en la plataforma quan el mode d'impressió ho correspongui. Aquests sempre es mostren a la part dreta de la superfície d'impressió i no es poden seleccionar, simplement copien els moviments dels originals. Les seves dades geomètriques van lligades a les del CuraSceneNode original, però corregint la seva posició en l'eix X.
- **Platform:** Representa la plataforma d'impressió de la impressora. Gestiona el canvi de plataforma quan es canvia de model d'impressora.
- **BuildVolume:** És l'encarregat de mostrar l'àrea d'impressió i les àrees que estan deshabilitades. També realitza els càlculs per a saber si els models estan dins d'aquesta àrea o no.
- **Camera:** Gestiona la perspectiva i el punt des d'on es veu l'escena.
- **ToolHandle:** Representa l'eina que es mostra sobre el model quan se li vol realitzar una operació de translació, escalat, rotació o mirall. Per tant té quatre subclasses que no es mostren en el diagrama perquè no són rellevants per al desenvolupament d'aquesta funcionalitat.

El PrintModeManager té una llista amb tots els elements del tipus DuplicatedNode i quan hi ha canvis en el mode d'impressió gestiona si s'han de mostrar en l'escena o no. També calcula el desplaçament en l'eix X en el que s'han de moure els elements del tipus CuraSceneNode que hi ha sobre la plataforma, ja que quan el mode d'impressió és *Mirror* o *Duplication* aquests han d'estar a la meitat esquerra de la superfície d'impressió.

### 7.3.2 Forçar assignació d'extrusors

Quan es canvia a mode d'impressió *Mirror* o *Duplication* els elements del tipus CuraSceneNode es mostren a la part esquerra de la superfície d'impressió i els del tipus DuplicatedNode es mostren a la part dreta. Per tant s'ha de forçar que els de la part esquerra estiguin assignats a l'extrusor esquerre i els de la dreta al dret. Si no fós així hi hauria col·lisions entre els capçals d'impressió, ja que s'haurien de creuar però no poden perquè es mouen sobre un mateix eix.

### 7.3.3 Restaurar el material de l'extrusor

Es pot donar el cas que cada extrusor tingui assignat un material diferent, però al canviar a mode d'impressió *Mirror* o *Duplication* els dos extrusors passen a tenir els mateixos valors i només està actiu l'extrusor de l'esquerre. Per recuperar el material que hi havia assignat originalment quan tornem al mode *Regular*, el PrintModeManager guarda el material antic en un dels fitxers on es guarda la informació de la sessió quan es tanca *Cura*. Així sempre que tornem al mode *Regular* podem recuperar el material que s'estava utilitzant a l'extrusor dret.

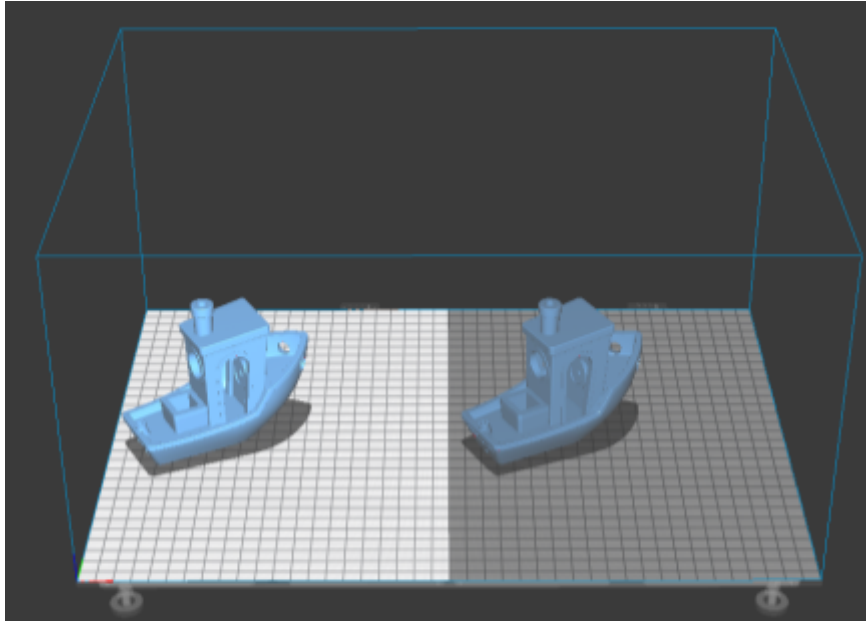
### 7.3.4 Resoldre els paràmetres conflictius

Els modes d'impressió *Mirror* i *Duplication* impliquen que els models que es volen imprimir només podran ser d'un extrusor, és a dir, tots els models originals s'imprimiran amb el capçal esquerre i els duplicats amb el dret. Alguns dels paràmetres d'impressió poden implicar que s'hagi d'usar en un mateix model els dos capçals, per tant a aquests paràmetres els hi forcem el valor que ens interessa i els desactivem per a evitar conflictes.

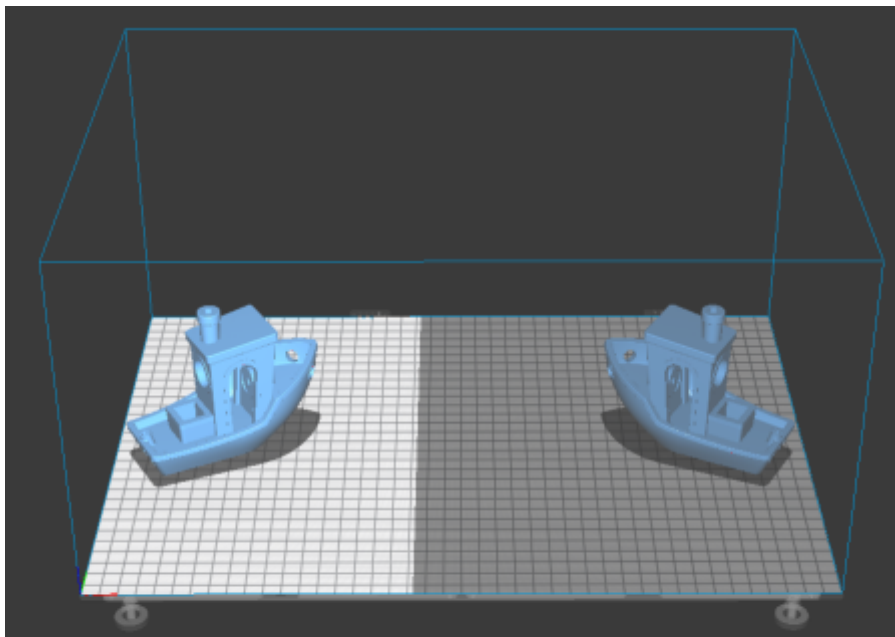
### 7.3.5 Calcular la superfície d'impressió

En els modes d'impressió *Mirror* i *Duplication* només podem realitzar accions sobre la part esquerra de la superfície d'impressió, ja que en la part dreta estan els models duplicats. Per tant la part dreta de la superfície d'impressió la tractem com a una àrea deshabilitada, tot i que s'hi mostren els models duplicats i s'hi acabarà imprimint. El PrintModeManager calcular el valor d'aquesta àrea en cada cas. Quan el mode d'impressió sigui *Duplication* aquesta àrea serà exactament la metitat de la superfície. En el cas del *Mirror* se li ha de sumar el tamany del capçal d'impressió, ja que si el model estigués situat al centre els dos capçals xocarien.

### 7.3.6 Resultat final



Duplication



Mirror

## 7.4 Connectivitat USB

El *Cura* original desenvolupat per *Ultimaker* ja inclou la funcionalitat de connectar la impressora mitjançant un cable USB, sempre i quan l'electrònica d'aquesta sigui una placa basada en arduino. Aquest és el cas de les impressores de BCN3D, però el comportament al connectar-les no era l'esperat i per això ha estat necessari modificar aquesta funcionalitat per a adaptar-se a les impressores de BCN3D.

Les classes de python que principalment es veuen implicades en aquesta funcionalitat són les que veiem en el diagrama.

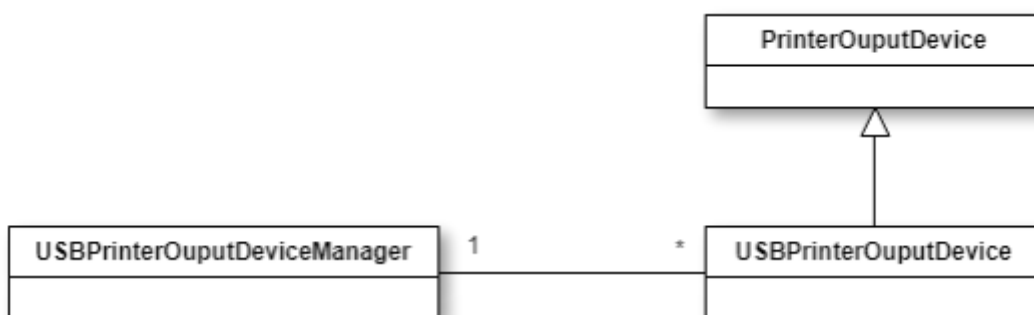


Diagrama de classes connexió USB

- **USBPrinterOutputDeviceManager:** Aquesta classe gestiona les impressores que s'han connectat per USB. Crea un *thread* que va comprovant si s'ha connectat o desconnectat alguna impressora a algun dels ports USB de l'ordinador. En el cas de que s'hagi connectat crea una nova instància de la classe USBPrinterOutputDevice.
- **PrinterOutputDevice:** És una classe que representa un dispositiu per a comunicar-se amb la impressora. Cada tipus de comunicació hauria de ser subclasse de PrinterOutputDevice i implementar els seus mètodes. Un exemple seria una subclasse per a comunicar-se a per wifi o com és el cas de les impressores de BCN3D via USB.
- **USBPrinterOutputDevice:** És una subclasse de PrinterOutputDevice i implementa els seus mètodes que no estan implementats. En aquesta classe es realitza la connexió amb la impressora, la comunicació i la gestió per a imprimir.

Per a iniciar la comunicació sèrie amb la impressora necessitem saber en quin port està connectada i quina és la velocitat de transmissió. El *Cura* original detectava la velocitat de transmissió de manera automàtica, ja que s'hi poden connectar varis models d'impressora. Aquesta manera de detectar la velocitat de transmissió automàticament és més lenta que si ja es coneix i per tant es realitza la connexió directament. Per això com que l'adaptació de

Cura que volem és específica per a les impressores de BCN3D i ja coneixem la seva velocitat de transmissió, farem la connexió directament utilitzant el mòdul *serial* de python.

Un cop estem connectats amb la impressora i s'ha obert la comunicació s'inicia un *thread* que serà on s'anirà llegint si hem rebut alguna resposta de la impressora.

### 7.4.1 Impressió per USB

Aquesta funcionalitat s'ha hagut de refer ja que de la manera com estava implementada la impressió era molt lenta. El problema estava en el protocol de comunicació. Un exemple senzill seria el següent:

```
while(connected):
    line = readLine() # Llegeix el que ha arribat de la impressora
    if line == "ok":
        sendNextGcodeLine() # envia la següent instrucció a la impressora
```

El problema que té aquest protocol de comunicació és que mentre la impressora contesta amb el missatge "ok" i s'envia la nova instrucció, si la impressora ja ha realitzat el moviment anterior estarà esperant a que arribi el nou, i per tant s'anirà parant fent que la impressió sigui més lenta i de pitjor qualitat. Però tampoc se li poden enviar instruccions massa ràpid, ja que no tindria capacitat suficient per a processar-les totes i algunes es perdrien pel camí. Per això s'ha modificat el protocol de comunicació pel següent:

```
buffer_size = 240
bytes_sent = 0
buffer = Queue() # cua per a les instruccions que estan al buffer
queue = Queue() # cua per a les instruccions que no caben al buffer
while connected:
    line = readLine() # Llegeix el que ha arribat de la impressora
    if line == "ok":
        bytes_sent -= len(buffer.get()) # restem dels bytes enviats i buidem del buffer
        if not queue.isEmpty():
            sendCommand(queue.get())
        else:
            sendCommand(next_instruction) # envia la següent instrucció a la impressora

def sendCommand(cmd):
    if bytes_sent + len(cmd) < buffer_size:
        send(cmd)
        bytes_sent += len(cmd)
        buffer.put(cmd)
    else:
        queue.put(cmd)
```



Aquest nou protocol ens permet anar enviant instruccions a la impressora encara que aquesta no les hagi acabat d'executar, així mai deixarà d'imprimir ja que sempre tindrà instruccions en cua. Per a fer això es necessari conèixer quin és el tamany del *buffer* de comunicació, ja que sinó es podrien perdre algunes de les instruccions que s'envien.

Quan s'està imprimint és important que totes les instruccions s'enviïn en ordre. En el fragment de codi anterior es pot donar el cas que quan s'intenta enviar una instrucció de la cua aquesta no hi càpiga al *buffer* i per tant es tornarà a posar a la cua, però l'ordre haurà canviat. Per a evitar-ho s'ha fet la següent modificació sobre la funció `sendCommand`:

```
def sendCommand(cmd, from_queue):
    if bytes_sent + len(cmd) < buffer_size:
        send(cmd)
        bytes_sent += len(cmd)
        buffer.put(cmd)
    elif from_queue:
        return
    else:
        queue.put(cmd)
```

Quan es crida la funció `sendCommand` passem un altre paràmetre que ens indica si la instrucció ve de la cua. En cas de que així sigui no la posarem de nou a la cua, però tampoc la treurem. Per a mantenir l'ordre volem que aquesta sigui la següent instrucció que s'envii.

Quan s'està imprimint per USB, cada instrucció que s'envia a la impressora se li afegeix el número de línia. D'aquesta manera el *firmware* pot comprovar que les instruccions s'estiguin executant en ordre. En cas que durant la comunicació es perdés alguna instrucció pel camí el *firmware* ho detectaria i ens tornaria un missatge dient que li tornem a enviar la instrucció amb aquell número de línia. Quan hi ha hagut un error d'aquest tipus, per a evitar que es propagui, buidarem les cues i tornarem a posar a zero el número de bytes enviats.

### 7.4.2 Actualització de *firmware*

Aquesta funcionalitat és molt important ja que contínuament es van fent millores sobre el *firmware* de les impressores i molts cops és per solucionar errors. Per això és important que l'usuari pugui realitzar aquesta actualització de forma fàcil i còmode.

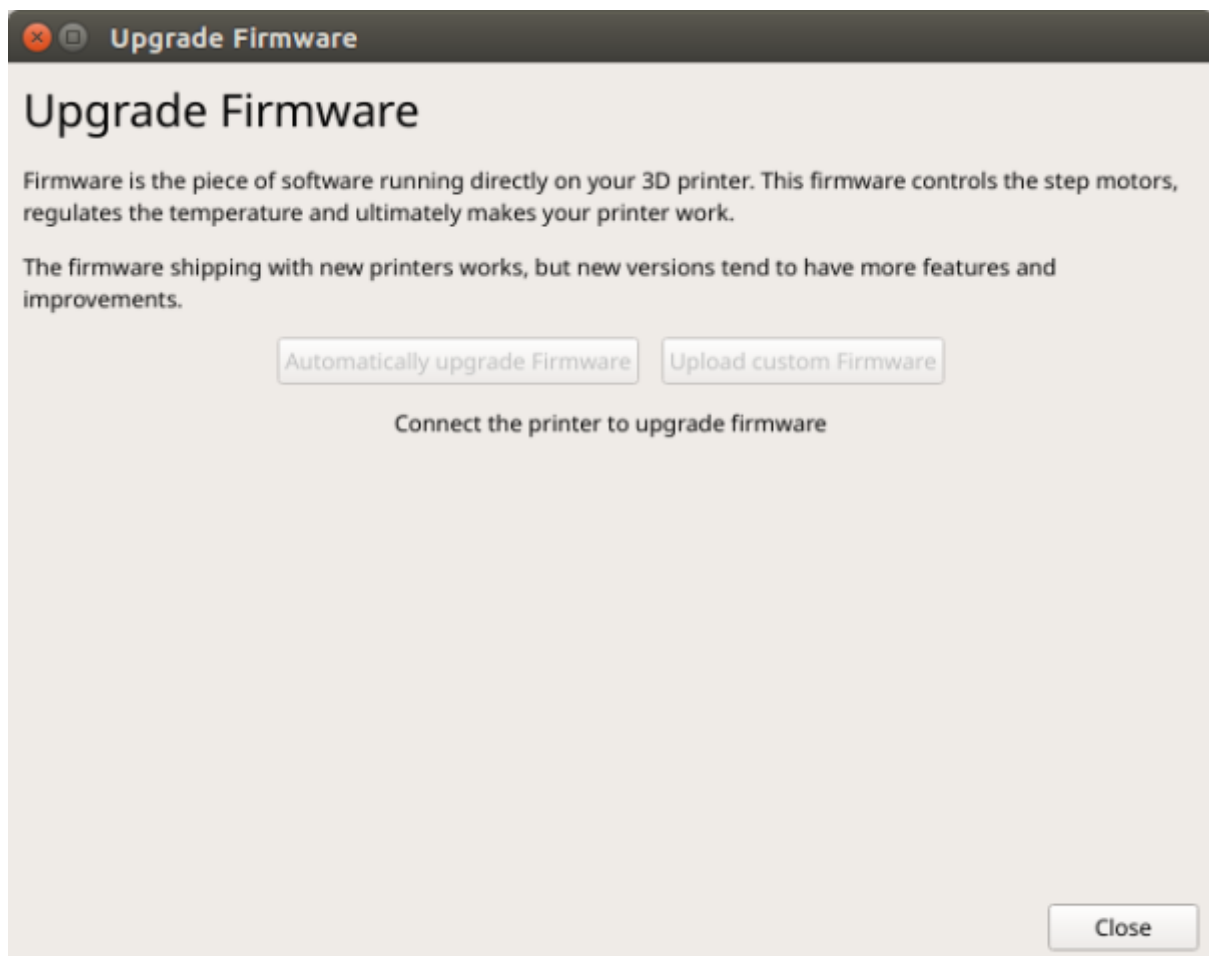
El *Cura* original ja incorpora aquesta funcionalitat. Hem pogut aprofitar la interfície gràfica, tot i que hem afegit algunes modificacions, el protocol per a carregar el *firmware* a l'electrònica de la impressora i carregar el *firmware* des d'un arxiu local. Però la funcionalitat que ens interessa a nosaltres és la de carregar el *firmware* de manera automàtica. L'usuari

només ha de clicar un botó, mentre la impressora està connectada per USB, i el *firmware* s'actualitza.

El *firmware* només el volem actualitzar si hi ha una versió més nova que la que té la nostre impressora, per tant necessitem saber l'última versió i la versió de la impressora que està connectada.

Quan s'estableix la connexió amb la impressora connectada per USB el primer missatge que envia la impressora és el número de versió. Aquest número el guardem per a fer-lo servir en cas d'actualitzar el *firmware*. En algunes versions del *firmware* el comportament no és aquest, i per saber la versió és necessari enviar una instrucció a la impressora.

Per saber la versió de l'última versió de *firmware* el que fem és utilitzar la API de *GitHub*. Quan surt una nova versió de *firmware*, aquesta es penja a *GitHub* i gràcies a la API podem consultar quina és l'última que s'ha penjat. Aleshores només cal comparar les versions i en el cas que la de *GitHub* sigui més nova descarregar-la. Si l'usuari té una versió beta, encara que sigui més nova, es descarregarà igualment la de *GitHub*. Per tant, per a actualitzar-se el *firmware* de manera automàtica és necessari disposar de connexió a internet.



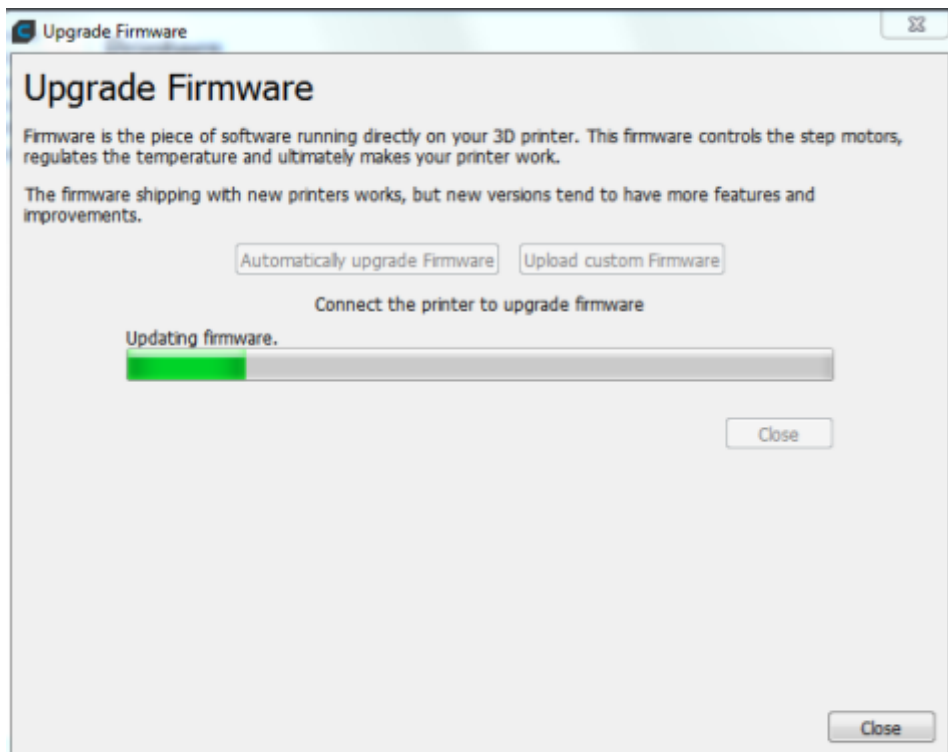
Pantalla d'actualització de firmware sense impressora connectada

Quan la impressora no està connectada l'opció d'actualitzar està desactivada, pero com veiem a la següent imatge, quan connectem la impressora els botons s'activen i es mostra la versió actual i l'última versió del *firmware*.



Pantalla d'actualització de firmware amb impressora connectada

Al clicar el botó de “Automatically upgrade Firmware” apareix una barra mostrant el progrés.



Pantalla procés actualització de firmware

### 7.4.3 Control de la impressora

En aquesta funcionalitat el que es vol és poder controlar certes funcions de la impressora des d'un ordinador quan aquesta està connectada per USB. Es vol poder moure els capçals d'impressió, tant en l'eix X com en l'eix Y. També es vol poder pujar i baixar la plataforma d'impressió. Les altres funcionalitats estan relacionades amb la temperatura. Escalfar els *hotends* i la plataforma d'impressió.

El *hotend* és el component que està situat en els capçals d'impressió i són els encarregats d'escalfar el material per a que surti en forma de fil.

Per últim es vol poder extruir material, és a dir, que surti material pels capçals d'impressió. Això es vol fer perquè pot ser que hi hagi brutícia o alguna imperfecció i d'aquesta manera ens assegurem de treure-la abans de començar una impressió.

Tots aquests controls no estan disponibles quan s'està duent a terme una impressió.

A la imatge veiem la barra lateral que es mostra quan estem a la pestanya dels controls per a la impressora.

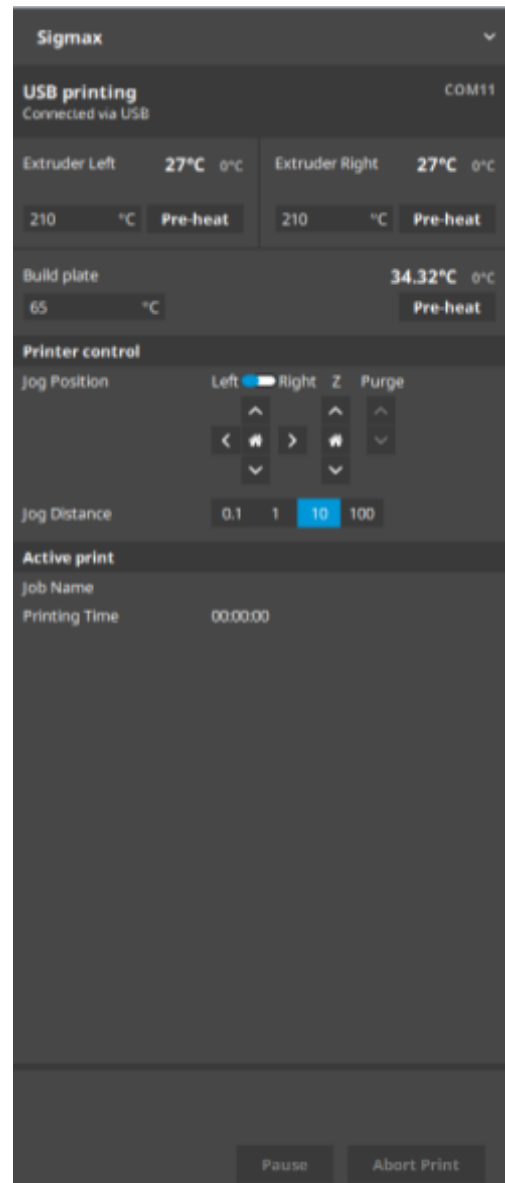
Podem triar a quina temperatura volem escalfar els extrusors i la plataforma d'impressió. La temperatura que es veu en el número gran és la actual. La del número petit gris és la objectiu. Com que no s'ha donat l'ordre d'escalfar és 0 °C.

Més avall hi ha els controls per a moure els capçals i la plataforma. En primer lloc hi ha un selector per canviar entre el capçal esquerre i el dret. Les quatre fletxes són per moure els capçals i les que estan sota de Z són per moure la plataforma. Al centre veiem un botó amb l'icona d'una casa. La seva funció és la de tornar a la posició inicial.

Les dues fletxes que estan desactivades serveixen per extruir material. A la imatge estan desactivades perquè els *hotends* no estan suficientment calents com per a fer-ho.

Podem seleccionar quina distància vole que es recorri en cada moviment (0.1, 1, 10 o 100) en mil·límetres.

Els dos botons que hi ha a la part inferior estan desactivats perquè no hi ha cap impressió activa. Ens permeten pausar la impressió, per a després reanudar-la, o cancel·lar-la.



Controls USB

## 7.5 Gestió dels executables

Durant el desenvolupament del projecte, quan el *software* que s'estava desenvolupant ha estat en un punt estable com per a treure un versió al públic s'han generat els executables per a que els usuaris el poguessin provar i així podíem anar rebent *feedback* i arreglant errors que no havíem detectat.

Per a generar els executables hem fet servir un projecte desenvolupat per *Ultimaker* que agafa tots els projectes necessaris de *GitHub*. Per tant hem hagut de modificar aquest projecte per a que agafi els projectes dels nostres repositoris de *GitHub*.

Per a que el *software* sigui accessible per a totes les plataformes hem hagut de crear els executables des d'un ordinador windows, des d'un mac i des d'un linux.

L'executable de windows es genera utilitzant NSIS, que permet crear un instal·lador propi de windows.

En el cas de mac es crea un arxiu amb extensió “.app”.

Per a linux crea un arxiu *AppImage*. Aquest arxiu pot ser executat en qualsevol sistema linux.

## 8 Conclusions i treball futur

Després d'haver acabat el projecte puc dir que estic molt satisfet amb la feina feta i sobretot per l'experiència viscuda i tot el que m'ha aportat en la meva formació. Ha estat el primer cop que he hagut de desenvolupar un projecte d'aquestes dimensions.

El més gratificant ha estat tenir l'oportunitat de realitzar-lo en una empresa que de veritat el necessitava i que l'hagin pogut utilitzar els usuaris dels seus productes. Veure que cada cop que sortia una nova versió del producte més de 1500 d'usuaris se'l descarregaven feia una mica de respecte. Això m'ha fet veure l'impacte que pot tenir un producte un cop està al mercat i que és molt important assegurar-se que aquest surt en un estat òptim.

En un futur m'agradaria continuar desenvolupant aquest producte ja que encara hi ha moltes millores per a fer. En aquest projecte s'ha desenvolupat la connexió per USB, però m'agradaria poder desenvolupar la connexió wifi, ja que és un tipus de connexió que s'adapta més per a les necessitats tecnològiques que hi ha actualment.

Una de les parts més importants que no ha entrat a l'abast d'aquest projecte és el motor que fa el *slicing*. És la part que té una influència més gran en la qualitat de la impressió i ja que en aquest projecte no s'ha tocat en un futur s'hauria d'intentar millorar.

## 9 Referències

### 9.1 Referències citades

1. “BCN3D.” [Online]. Available: <http://www.bcn3dtechnologies.com>. [Accessed: 2018].
2. “Fundació CIM.” [Online]. Available: <https://www.fundaciocim.org>. [Accessed: 2018].
3. “Fused Deposition Modeling: Most Common 3D Printing Method.” [Online]. Available: <https://www.livescience.com/39810-fused-deposition-modeling.html>. [Accessed: 2018].
4. “What is Slicing Software, and what does it do?” [Online]. Available: <https://www.goprint3d.co.uk/blog/what-is-slicing-software-and-what-does-it-do/>. [Accessed: 2018].
5. “Ultimaker Cura 3D Printing Software.” [Online]. Available: <https://ultimaker.com/en/products/ultimaker-cura-software>. [Accessed: 2018].
6. “Asana.” [Online]. Available: <https://asana.com/>. [Accessed: 2018].
7. “What is Scrum?” [Online]. Available: <https://www.scrum.org/resources/what-is-scrum>. [Accessed: 2018].
8. “GitHub.” [Online]. Available: <https://github.com/>. [Accessed: 2018].
9. “Ultimaker: 3D Printers.” [Online]. Available: <https://ultimaker.com/>. [Accessed: 2018].

### 9.2 Altres referències

1. “What is 3D printing? How does 3D printing work? Learn How to 3D Print.” [Online]. Available: <https://3dprinting.com/what-is-3d-printing/>. [Accessed: 2018].
2. “The 3D Printing Process: 7 Steps.” [Online]. Available: <http://www.instructables.com/id/The-3D-Printing-Process/>. [Accessed: 2018].
3. “Types of 3D printers or 3D printing technologies overview | 3D Printing from scratch.” [Online]. Available: <http://3dprintingfromscratch.com/common/types-of-3d-printers-or-3d-printing-technologies-overview/>. [2018].

4. "Impresión 3D FDM – Integral 3D Printing." [Online]. Available:  
<http://integral3dprinting.com/tecnologias-de-impresion-3d/impresion-3d-fdm/>.  
[Accessed: 2018].
5. "FDM technology: The advantages of FDM printing | dddrop." [Online]. Available:  
<https://www.dddrop.com/fdm-technology/>. [2018].